UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/649,903	08/26/2003	Mahesh A. Ramchandani	5150-77400	5752

7590 06/11/2007

Jeffrey C. Hood
Meyertons, Hood, Kivlin,
Kowert & Goetzel PC
P.O. Box 398
Austin, TX 78767

EXAMINER	
MITCHELL, JASON D	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
06/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/649,903	RAMCHANDANI, MAHESH A.	
	Examiner	Art Unit	
	Jason Mitchell	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *76-114* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *76-114* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 76-114 are pending in this application.

Response to Arguments

2. Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. **Claims 86 and 103 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.**

The claims contain the trademark/trade name "ActiveX". Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the

present case, the trademark/trade name is used to identify/describe 'a control' and, accordingly, the identification/description is indefinite.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 76-90 and 92-95 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,401,220 to Grey et al. (Grey) in view of US 5,485,617 to Stutz et al. (Stutz)**

7. **Regarding Claim 76:** Grey discloses a computer-implemented method for displaying information regarding a test executive sequence, wherein the test executive sequence includes a plurality of steps, the method comprising:

including a GUI element in a run-time operator interface application, wherein the GUI element is operable to display information (Fig. 4 see the 'Main' tab, 'Step' column);

including a control in the run-time operator interface application, wherein the control includes pre-existing first functionality for determining the steps in the test executive sequence (col. 4, lines 4-5 "The TestStand Engine automatically determines the type being loaded");

configuring a binding between the GUI element and the control (col. 3, lines 16-18 "The sequence editor ... interface[s] to the test executive engine"); and

executing the run-time operator interface application, wherein the control is operable to automatically determine the steps in the test executive sequence during execution of the run-time operator interface application (col. 4, lines 4-5 "The TestStand Engine automatically determines the type being loaded"), wherein the binding between the GUI element and the control causes the control to display at least a subset of the steps in the GUI element in response to the control determining the steps (Fig. 4 see the 'Main' tab, 'Step' column).

8. Grey does not disclose including the GUI element and control in the run-time operator interface in response to user input.

9. Stutz teaches that including GUI elements and controls in a run-time operator interface is done in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display").

10. It would have been obvious to one of ordinary skill in the art at the time the invention was made to develop the run-time operator interface disclosed in Grey using the methods taught in Stutz (col. 10, lines 46-48) because Stutz provides "an improved method ... for dynamically generating object connections" (col. 8, lines 14-17).

11. **Regarding Claim 77:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the control also includes pre-existing functionality for formatting the at least a subset of the steps in the test executive sequence into a displayable list of the at least a subset of the steps (col. 31, lines 31-33 "specify a name, description, and comment for the step type. The user also can specify an icon and an module adapter");

wherein said displaying the at least a subset of the steps in the GUI element comprises displaying the list of the at least a subset of the steps (Fig. 4 see the 'Main' tab, 'Step' column).

12. **Regarding Claim 78:** The rejection of claim 77 is incorporated; further Grey discloses:

wherein in performing said formatting the at least a subset of the steps in the test executive sequence into the displayable list, the control is operable to:

determine information regarding each of the at least a subset of the steps in the test executive sequence; and format the information for display in the GUI element;

wherein the displayable list includes the formatted information for each of the steps in the at least a subset of the steps (Fig. 4, see 'Main' tab).

13. **Regarding Claim 79:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the test executive sequence is stored in a sequence file (col. 5, lines 53-54 "a test sequence file");

wherein in automatically determining the steps in the test executive sequence, the control is operable to automatically obtain information from the sequence file regarding the test executive sequence and determine the steps based on the information obtained from the sequence file (col. 4, lines 1-5 "the user loads a file ... The TestStand Engine automatically determines the type being loaded").

14. **Regarding Claim 80:** The rejection of claim 76 is incorporated; further Grey discloses:

configuring the control in response to configuration user input after said including the control in the run-time operator interface application (col. 31, lines 31-33 "specify a name, description, and comment for the step type. The user also can specify an icon and an module adapter"), wherein the configuration user input specifies an appearance for the displayed steps, wherein configuring the control enables the control to cause the steps to be displayed in the GUI element with the specified appearance (Fig. 4, see 'Main' tab).

15. **Regarding Claim 81:** The rejection of claim 80 is incorporated; further Grey discloses:

wherein the configuration user input specifies one or more properties regarding a plurality of columns to display in the GUI element (col. 31, lines 31-33 "specify a name,

description, and comment for the step type. The user also can specify an icon and an module adapter"); wherein configuring the control enables the control to cause information for each displayed step to be displayed in the GUI element in the plurality of columns according to the one or more specified properties (Fig. 4, see 'Main' tab).

16. **Regarding Claim 82:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the GUI element comprises a first GUI element;

wherein the method further comprises:

including a second GUI element in the run-time operator interface application in response to user input (col. 4, lines 4-5 "the user loading the file"; Note that those of ordinary skill in the art would have recognized this action is preformed through a GUI element.); and

configuring a binding between the second GUI element and the control (col. 3, lines 16-19 "the operator interface programs interface to the test executive engine");

wherein the second GUI element is operable to receive user input during execution of the run-time operator interface application (col. 4, lines 4-5 "the user loading the file");

wherein said configuring the binding between the second GUI element and the control enables the control to automatically determine the steps in the test executive sequence in response to the user input received to the second GUI element (col. 4,

lines 4-5 "In response to the user loading the file, the TestStand Engine automatically determines the type being loaded");

wherein said configuring the binding between the first GUI element and the control enables the first GUI element to automatically display the at least a subset of the steps in response to the user input received to the second GUI element (Fig. 4 see the 'Main' tab, 'Step' column).

17. **Regarding Claim 83:** The rejection of claim 76 is incorporated; further Grey does not disclose including the control in the run-time operator interface application or that configuring the binding between the GUI element removes a need for a user to create program code for providing these functionalities.

18. Stutz teaches including a control in the run-time operator interface application enables a user to configure the run-time operator interface application to perform a first functionality without requiring the user to create program code (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected"); and

configuring a binding between a GUI element and the control enables the user to configure the run-time operator interface application to automatically perform display the results of said first functionality without requiring the user to create program code for displaying the results (col. 11, lines 5-11 "Using the various commands provided by the buttons in the command area 502").

19. **Regarding Claim 84:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the test executive sequence is operable to perform one or more tests on one or more units under test (UUTs) (col. 4, lines 47-48 "A sequence comprises a series of steps, wherein a step is typically a test preformed on an instrument.").

20. **Regarding Claim 85:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the test executive sequence is associated with a test executive environment (col. 13, lines 32-33 "The TestStand Test Executive Engine 220 is used for creating, editing, executing, and debugging sequences.");

wherein the control is operable to call the test executive environment during execution of the run-time operator interface application to determine the steps in the test executive sequence (col. 13, lines 39-41 "The user can call the Engine API from any programming environment").

21. **Regarding Claim 86:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the control comprises an ActiveX component (col. 3, lines 30-33 "The TestStand Engine exports an ActiveX automation API").

22. **Regarding Claim 87:** The rejection of claim 76 is incorporated; further Grey discloses:

wherein the GUI element appears on a graphical user interface of the run-time operator interface application during execution of the run-time operator interface application (Fig. 4 see the 'Main' tab, 'Step' column);

wherein the GUI element is operable to display the steps on the graphical user interface during execution of the run-time operator interface application (Fig. 4 see the 'Main' tab, 'Step' column).

Regarding Claim 88: The rejection of claim 87 is incorporated; further, while not explicitly stated, it is clear from Grey's disclosure that the control (col. 3, lines 30-33 "The TestStand Engine") does not appear on the graphical user interface of the run-time operator interface application during execution of the run-time operator interface application.

23. **Regarding Claim 89:** The rejection of claim 76 is incorporated; further Grey does not disclose the control is a pre-existing control provided by an application development environment.

24. Stutz teaches a pre-existing control provided by an application development environment (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected").

25. **Regarding Claim 90:** The rejection of claim 89 is incorporated; further Grey does not explicitly disclose installing an application development environment on a computer system.

26. Stutz discloses both the application development environment and the control (col. 10, lines 42-45 "visual programming environment ... list of predefined components") "implemented on a computer system" (col. 9, lines 15-21). Accordingly, both the application development environment and the control must have been installed on the computer system

27. **Regarding Claim 92:** The rejection of claim 76 is incorporated; further Grey does not disclose the configuring the binding between the GUI element and the control comprises performing one or more calls during execution of the run-time operator interface application.

28. Stutz discloses said configuring the binding between the GUI element and the control comprises performing one or more calls to bind the GUI element to the control during execution of the test executive application (col. 15, lines 49-52 "The function SetUpConnection ... connects the appropriate notification interface").

29. **Regarding Claim 93:** The rejection of claim 76 is incorporated; further Grey does not disclose configuring the binding between the GUI element and the control is preformed in response to user input.

30. Stutz teaches configuring a binding between a GUI element and a control is performed in response to receiving user input to a graphical user interface to specify the binding between the GUI element and the control (col. 11, lines 5-11 "Using the various commands provided by the buttons in the command area 502").

31. **Regarding Claims 94, and 95:** Grey discloses a computer-implemented method for displaying a report for a test executive sequence execution, the method comprising:

including a GUI element in a run-time operator interface application, wherein the GUI element is operable to display information (Fig. 50, see the 'Context' tab, 'ResultList' node; also see the 'Report' tab);

including a control in the run-time operator interface application, wherein the control includes pre-existing first functionality for generating a report for an execution of the test executive sequence (col. 8, lines 9-10 "The TestStand Engine operates to automatically collect the results of each step in the sequence during execution"; col. 56, lines 48-50 "TestReport ... to generate the contents of the test report");

configuring a binding between the GUI element and the control (col. 3, lines 16-18 "the operator interface programs interface to the test executive engine");

configuring the run-time operator interface application to invoke execution of the test executive sequence (col. 7, lines 18-24 "The user preferably begins execution of the process model through a graphical user interface"); and

executing the run-time operator interface application, wherein the run-time operator interface application executes to invoke execution of the test executive sequence (col. 7, lines 18-24 "execution of the process model"), wherein the execution of the test executive sequence produces one or more results (col. 8, lines 9-10 "The TestStand Engine operates to automatically collect the results of each step in the sequence during execution"), wherein the control is operable to automatically generate a report summarizing the one or more results of the execution of the test executive sequence in response to the execution of the test executive sequence (col. 56, lines 48-50 "TestReport ... to generate the contents of the test report"), wherein the binding between the GUI element and the control causes the report to be displayed by the GUI element in response to the control generating the report (Fig. 50, see the 'Context' tab, 'ResultList' node; also see the 'Report' tab).

32. Grey does not disclose including the GUI element and control in the run-time operator interface in response to user input.

33. Stutz teaches that including GUI elements and controls in a run-time operator interface is done in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display").

34. It would have been obvious to one of ordinary skill in the art at the time the invention was made to develop the run-time operator interface disclosed in Grey using the methods taught in Stutz (col. 10, lines 46-48) because Stutz provides "an improved method ... for dynamically generating object connections" (col. 8, lines 14-17).

35. **Claim 91 is rejected under 35 U.S.C. 103(a) as being unpatentable over US 6,401,220 to Grey et al. (Grey) in view of US 5,485,617 to Stutz et al. (Stutz) and further in view of US 6,718,534 to Carter et al. (Carter).**

36. **Regarding Claim 91:** The rejection of claim 89 is incorporated; further the Grey-Stutz combination does not disclose installing the control on the computer system after installing the application development environment.

37. Carter teaches installing a control after installing an application development environment (col. 6, lines 3-5 "The user can import a control from an external source").

38. It would have been obvious to a person of ordinary skill in the art at the time of the invention to combine the teachings of the Stutz-Grey combination in order to avoid "replication of the control programmability function" and the associated Duplication of work and increased cost. (Carter col. 1, lines 48-52)

39. **Claims 96-114 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 5,485,617 to Stutz et al. (Stutz) in view of US 6,401,220 to Grey et al. (Grey).**

40. **Regarding Claim 96:** Stutz discloses a computer-implemented method for creating a run-time operator interface application, the method comprising:

including a GUI element in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display");

including a control in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"), wherein the control includes pre-existing first functionality (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected"; col. 11, lines 11-17 "code for updating the list of files"); and

configuring a binding between the GUI element and the control (col. 11, lines 11-17 "the output port 516 of the open file dialog box object 503 has been connected to the input port 517 of the code object 504");

wherein the GUI element is operable to receive user input during execution of the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box operates by responding to ... events raised from user input");

wherein said configuring the binding between the GUI element and the control enables the control to automatically invoke execution of the first functionality in response to the user input received to the GUI element (col. 11, lines 42-52 "Upon receiving this selection event ... The code object 504 ... sends an updated file list to the multiple selection list box object 509").

41. Stutz does not disclose the run-time operator interface application controls execution of a test executive sequence or a control providing functionality for invoking execution of the test executive sequence.

42. Grey discloses an operator interface application for controlling execution of a test executive sequence (col. 2, lines 61-64 "the TestStand test executive software"); and a control providing functionality for invoking execution of the test executive sequence (col. 3, lines 23-25 "The TestStand Engine executes sequences").

43. It would have been obvious to one of ordinary skill in the art at the time the invention was made to create an application for controlling execution of a test executive sequence (Grey, col. 2, lines 61-64), including a control providing functionality for invoking execution of the test executive sequence (Grey col. 3, lines 23-25) using the development environment disclosed by Stutz because it provides "an improved method ... for dynamically generating object connections" (Stutz col. 8, lines 14-17).

44. **Regarding Claim 97:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein the GUI element comprises a first GUI element;

wherein the method further comprises:

including a second GUI element in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"); and

configuring a binding between the second GUI element and the control (col. 11, lines 15-17 "the input port 511 of the multiple selection list box object 509 has been connected to the output port 518 of the code object 503");

wherein said configuring the binding between the second GUI element and the control enables the control to automatically perform pre-existing second functionality during execution of the run-time operator interface application (col. 11, lines 42-52 "Upon receiving this selection event ... The code object 504 ... sends an updated file list to the multiple selection list box object 509").

45. Stutz does not disclose the control also includes pre-existing second functionality for displaying steps in the test executive sequence.

46. Grey teaches a control which includes functionality for displaying steps in the test executive sequence (col. 3, lines 14-15 "a sequence editor").

Patent Training Academy Lecture Hall - MDW207

47. **Regarding Claim 98:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein the control is a first control;

wherein the method further comprises including a second control in the run-time operator interface application in response to user input, wherein the second control includes pre-existing second functionality (col. 11, lines 11-15 "multiple selection list box object 509");

wherein said configuring the binding between the GUI element and the first control also enables the first control to invoke the second control to perform the second functionality (col. 11, lines 11-15 "code for updating the list of files shown in the multiple selection list box object 509").

48. **Regarding Claim 99:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein said configuring the binding between the GUI element and the control enables a user to configure the run-time operator interface application to invoke the first functionality in response to user input received to the GUI element without requiring the user to write program code to program the run-time operator interface application to respond to user input received to the GUI element (col. 11, lines 5-11 "Using the various commands provided by the buttons in the command area 502").

49. **Regarding Claim 100:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein said configuring the binding between the GUI element and the control enables a user to configure the run-time operator interface application to invoke execution of the first functionality in response to user input received to the GUI element without requiring the user to create program code to program the run-time operator interface to invoke execution of the test executive sequence (col. 11, lines 5-11 "Using the various commands provided by the buttons in the command area 502").

50. **Regarding Claim 101:** The rejection of claim 96 is incorporated; further Stutz does not disclose a control operable to invoke execution of a test executive sequence to perform one or more tests on one or more units under test.

51. Grey teaches a control is operable to invoke execution of the test executive sequence to perform one or more tests on one or more units under test (UUTs) (col. 3, lines 23-25 "The TestStand Engine executes sequences"; col. 4, lines 47-48 "A sequence comprises a series of steps, wherein a step is typically a test preformed on an instrument.").

52. **Regarding Claim 102:** The rejection of claim 96 is incorporated; further Stutz does not disclose a test executive sequence or a test executive environment.

53. Grey teaches the test executive sequence is associated with a test executive environment (col. 13, lines 32-33 "The TestStand Test Executive Engine 220 is used for creating, editing, executing, and debugging sequences."); and

a control operable to automatically call the test executive environment during execution of the run-time operator interface application to invoke execution of the test executive sequence (col. 13, lines 39-41 "The user can call the Engine API from any programming environment").

54. **Regarding Claim 103:** The rejection of claim 96 is incorporated; further Stutz does not disclose the control comprises an ActiveX control.

Grey teaches the control comprises an ActiveX control (col. 3, lines 30-32 "The TestStand Engine exports an ActiveX Automation API").

55. **Regarding Claim 104:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein the GUI element appears on a graphical user interface of the run-time operator interface application during execution of the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box");

wherein the GUI element is operable to receive user input to the graphical user interface during execution of the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box operates by responding to ... user input").

56. **Regarding Claim 105:** The rejection of claim 104 is incorporated; further Stutz discloses:

wherein the control does not appear on the graphical user interface of the run-time operator interface application during execution of the run-time operator interface application (Code Objects 504-507 in Fig. 5, the term "Code Object" itself as well as the fact that Code Objects 504-507 are depicted outside of Open File Dialog Box 503 indicates that the control does not appear on the graphical user interface).

57. **Regarding Claim 106:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein the control is a pre-existing control provided by an application development environment used to create the run-time operator interface application (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected").

58. **Regarding Claim 107:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein said configuring the binding between the GUI element and the control is performed in response to receiving user input to a graphical user interface to specify the binding between the GUI element and the control (col. 11, lines 5-11 "Using the various commands provided by the buttons in the command area 502").

59. **Regarding Claim 108:** The rejection of claim 96 is incorporated; further Stutz discloses:

executing the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box operates by responding to particular system events";);

receiving user input to the GUI element during execution of the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box operates by responding to ... events raised from user input"); and

the control invoking the first functionality in response to said receiving user input to the GUI element (col. 11, lines 42-52 "Upon receiving this selection event ... The code object 504 ... sends an updated file list to the multiple selection list box object 509").

60. **Regarding Claim 109:** The rejection of claim 96 is incorporated; further Stutz discloses:

wherein said including the GUI element in the run-time operator interface application in response to user input comprises displaying the GUI element on a graphical user interface of the run-time operator interface application in response to user input (col. 11, lines 37-40 "the open file dialog box operates by responding to ... events raised from user input").

Regarding Claim 110: The rejection of claim 96 is incorporated; further Stutz does not explicitly disclose configuring properties of the control after configuring the binding between the GUI element and the control.

Grey teaches configuring one or more properties of the control in response to user input after said configuring the binding between the GUI element and the control (col. 31, lines 31-33 "specify a name, description, and comment for the step type. The user also can specify an icon and an module adapter").

61. **Regarding Claim 111:** The rejection of claim 110 is incorporated; further Stutz does not disclose displaying a property panel for configuring the control.

Grey teaches displaying a property panel for configuring the control; and receiving user input to the property panel to configure the one or more properties of the control (col. 31, lines 31-33 "The General tab is used to specify a name").

Regarding Claim 112: The rejection of claim 96 is incorporated; further Stutz discloses:

wherein the GUI element comprises one or more of:

a button (Fig. 5, "Ok" button 510);

a text input element;

a check box;

a selection ring.

62. **Regarding Claim 113:** Stutz discloses a computer-implemented method for creating a run-time operator interface application, the method comprising:

including a GUI element in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display");

including a control in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"), wherein the control includes pre-existing first functionality (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected"; col. 11, lines 11-17 "code for updating the list of files"); and

configuring a binding between the GUI element and the control (col. 11, lines 11-17 "the output port 516 of the open file dialog box object 503 has been connected to the input port 517 of the code object 504");

wherein the GUI element is operable to receive user input during execution of the run-time operator interface application (col. 11, lines 37-40 "the open file dialog box operates by responding to ... events raised from user input");

wherein said configuring the binding between the GUI element and the control enables the control to perform the first functionality in response to the user input received to the GUI element (col. 11, lines 42-52 "Upon receiving this selection event ... The code object 504 ... sends an updated file list to the multiple selection list box object 509").

63. Stutz does not disclose the run-time operator interface application controls execution of a test executive sequence or that the control provides functionality for stopping execution of a test executive sequence.

64. Grey discloses an operator interface application for controlling execution of a test executive sequence (col. 2, lines 61-64 "the TestStand test executive software"); and a control providing functionality to automatically stop execution of the test executive sequence (col. 24, lines 1-4 "run-time operator interfaces 202 have commands that allow the user to stop execution").

65. It would have been obvious to one of ordinary skill in the art at the time the invention was made to create an application for controlling execution of a test executive sequence (Grey, col. 2, lines 61-64), including a control providing functionality to automatically stop execution of the test executive sequence (Grey col. 24, lines 1-4) using the development environment disclosed by Stutz because it provides "an improved method ... for dynamically generating object connections" (Stutz col. 8, lines 14-17).

66. **Regarding Claim 114:** Stutz discloses a computer-implemented method for creating a run-time operator interface application, the method comprising:

including a first GUI element in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"), wherein the first GUI element is operable to receive user input during execution of the run-time operator interface application (col. 11, lines 40-52 "the open file dialog box object 503");

including a second GUI element in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"), wherein the second GUI element is operable to display output during execution of the run-time operator interface application (col. 11, lines 40-52 "the multiple selection list box object 509");

including a control in the run-time operator interface application in response to user input (col. 10, lines 46-48 "specifies the visual components and their location on the display"), wherein the control includes pre-existing functionality (col. 10, lines 42-45 "a list of predefined components (objects) that can be interconnected"; col. 11, lines 11-17 "code for updating the list of files");

configuring a first binding between the first GUI element and the control, wherein the first binding enables the control to automatically invoke a dialog box in response to user input received to the first GUI element (col. 11, lines 42-52 "Upon receiving this selection event ... The code object 504"); and

configuring a second binding between the second GUI element and the control, wherein the second binding enables the control to automatically display information in the second GUI element in response to the user selecting the test executive sequence

(col. 11, lines 42-52 "The code object 504 ... sends an updated file list to the multiple selection list box object 509").

67. Stutz does not disclose the run-time operator interface application controls execution of a test executive sequence and does not disclose a control having functionality for selecting a test executive sequence or for displaying the steps of a test executive sequence.

68. Grey discloses an operator interface application for controlling execution of a test executive sequence (col. 2, lines 61-64 "the TestStand test executive software"); and a control providing functionality for selecting a test executive sequence (col. 6, lines 33-34 "the user ... selects a previously configured process model") and for displaying steps in the test executive sequence (col. 3, lines 14-15 "a sequence editor").

69. It would have been obvious to one of ordinary skill in the art at the time the invention was made to create an application for controlling execution of a test executive sequence (Grey, col. 2, lines 61-64), including a control providing functionality for selecting a test executive sequence (Grey col. 6, lines 33-34) and for displaying steps in the test executive sequence (Grey col. 3, lines 14-15) using the development environment disclosed by Stutz because it provides "an improved method ... for dynamically generating object connections" (Stutz col. 8, lines 14-17).

Conclusion

70. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jason Mitchell whose telephone number is (571) 272-3728. The examiner can normally be reached on Monday-Thursday and alternate Fridays 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jason Mitchell
5/31/07

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100